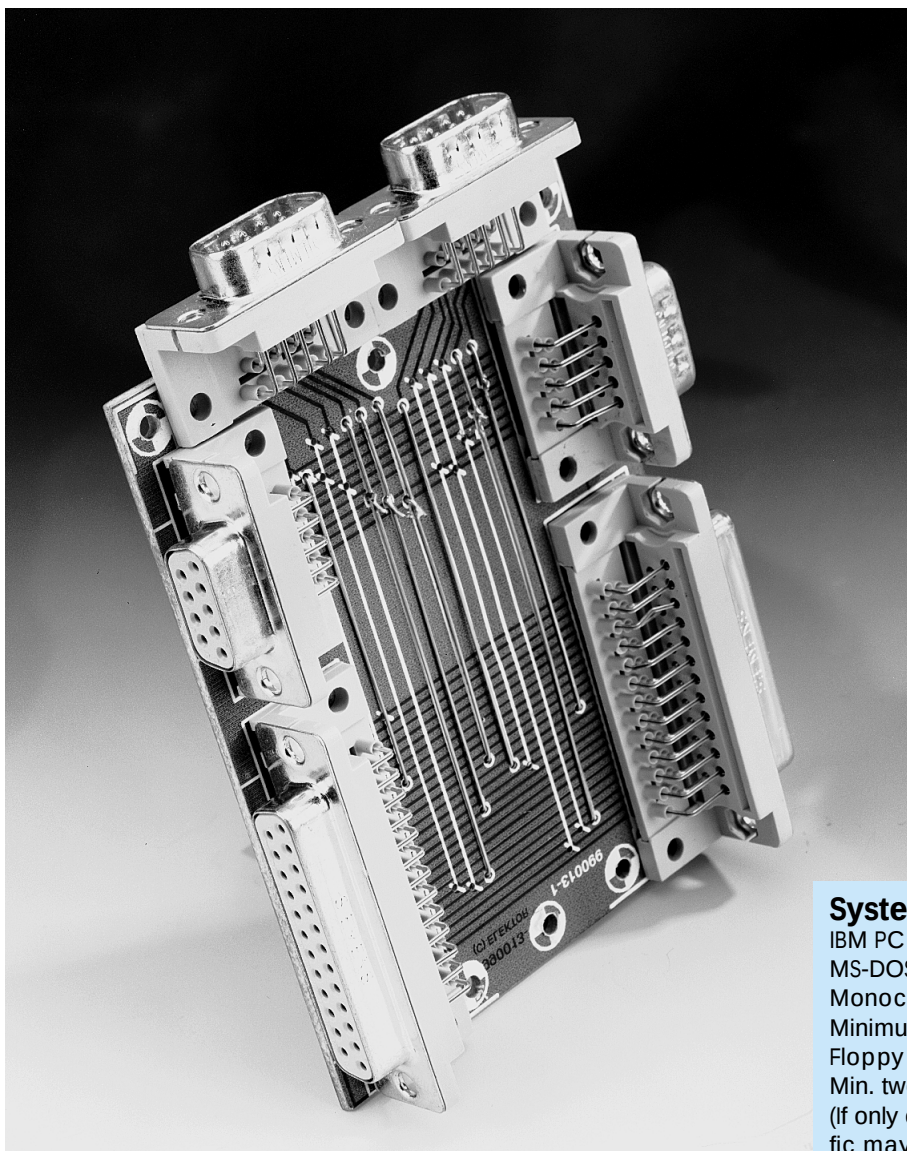


To many computer users, asynchronous serial communication will always remain something of a mystery. Does the link work or not? Do all the peripherals have the same settings, and does the right signal arrive on the right pin? These questions are easily answered by Com-Watch®. The system consists of an advanced DOS program and a simple passive adaptor.

In co-operation with CER International

COM-Watch®

problem sleuth for serial ports



Although the serial port is one of the most frequently used communication gateways in the modern PC, it also calls up the largest number of questions. Modem, mouse, graphics tablet, digital photo camera, personal organiser, you mention it, all this equipment is designed to exchange data with the PC by way of the serial port. The difficulty with this port is the ever-changing number of control signals employed: sometimes three, sometimes five, and then again eight! This should not be a problem, though, if only the transmitter and the receiver use the same communication settings.

The problem is aggravated by the fact that there is a difference between a DTE (data terminal equipment, for example, a PC or a printer), and a DCE (data communication equipment, for example, a modem). As a result, two computers that have to communicate with each other (i.e., two DTEs), have to be connected by a different cable than, say, a computer (DTE) and a modem (DCE).

Note, however, that there are still lots

System requirements

- IBM PC XT, AT or better
- MS-DOS 3.1 or later
- Monochrome or colour videocard
- Minimum 256 kBytes memory free
- Floppy disk drive (3.5 in.)
- Min. two free serial ports
- (If only one port is available, one-way data traffic may be monitored)

Com-Watch is a registered trademark of CER International b.v.

of pitfalls even if the right hardware (wires and connectors) is used. There is, for instance, the baud rate to take into account, the number of data bits (7 or 8), the number of stop bits and the parity check.

Considering all these potential problems, it is surprising to see that so many serial links work just fine.

The complexity of the matter is evident from the very fact that so many different systems have been developed to analyse serial connections, and resolve conflicts. COM-watch is a system developed by CER international specifically for this purpose. Using very little hardware and a DOS program, all traffic on an RS232 link may be scrutinised in great detail, allowing the user to detect the cause of the communication conflict, and, of course, eliminate the source of the problem.

CER International and *Elektor Electronics* jointly offer the program at a special reduced price. The associated hardware is simple to make, and you'll find it described further on in this article.

The approach

The basic operation of COM-Watch is illustrated in **Figure 1**. In principle, there are two ways of analysing a serial link: actively or passively. Com-Watch supports both, so that the system can really be called versatile.

The first approach that comes to mind is to observe the operation of an existing link. This is shown in **Figure 1a**. The method as shown employs passive analysis. Two apparatus communicate by means of a serial connection. Using a special adaptor (which will be reverted to below), the link is 'tapped'. A PC running the COM-Watch software then analyses the data traffic, and attempts to make sense of any errors that occur. Note that the PC used for fault analysis does not partake in the data traffic: it merely acts as an onlooker, i.e., its role is passive. The PC uses two serial ports (Channel 1 and Channel 2) to monitor the communication process: one for the data traffic from DCE to DTE, and for DTE-to-DCE traffic. Both channels have their own connection on the interface board to be described.

On the PC, serial ports COM1 through COM4 may be used for Channel 1 and Channel 2. For maximum functionality, two physical ports are needed. If only one port is available, then the channel to use for a particular set of signals may be found in **Table 1**. By changing over to another connector on the interface board, the entire

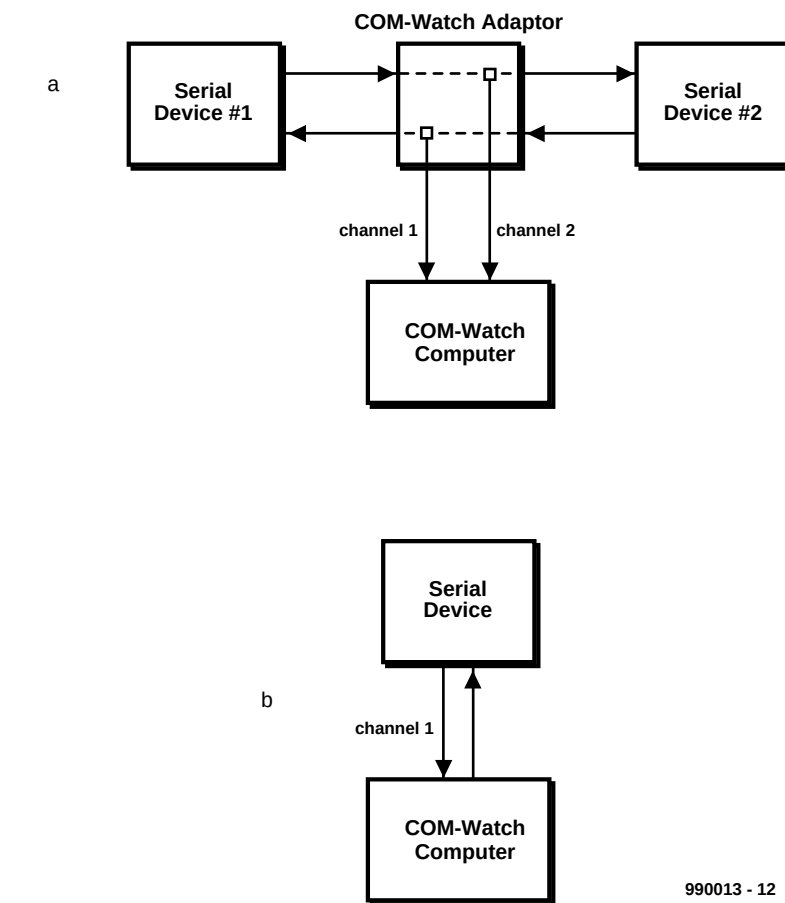


Figure 1. Using COM-Watch, data traffic may be analysed in two ways, passively (a) or actively (b). The passive approach requires a special adaptor.

communication process may be still analysed with just one extra action.

In passive mode, all data traffic is monitored, but not affected in any way. In true Sherlock Holmes tradition, Com-Watch tries to observe what is happening, and detect flaws without any of the 'suspects' noticing. On the display of the PC running the Com-Watch program appears a detailed analysis of the problem situation.

The active alternative

If you are having problems in getting a serial port to work at all, Com-Watch offers an active mode to help you sort it all out. Figure 2b shows how it works. Via its serial port the analysing computer communicates with the peripheral, and so collects as much information as possible. The display shows which data is sent by the PC, and which data is returned. The switching of several control signals is also easy to follow. This measurement method does not require specific hardware.

Data received by the computer is displayed by the software via Channel 1 (remote channel), while that sent by the computer to the peripheral

apparatus may be viewed on Channel 2 (local channel). The latter Channel is only implemented in software, and therefore does not require a physical serial communication port.

On paper, the approach used by Com-Watch may look complex. In practice, the process is self-evident. The internal context-sensitive Help function makes working with the software much easier.

A little hardware

To be able to analyse the connections sketched in Figure 2a, a special adaptor circuit has been developed. As shown in **Figure 2**, this is a very simple circuit. For do-it-yourself construction, a single-sided printed circuit board is available. Few of you will object to the one disadvantage of a single-sided board: having to fit a couple of wire links. The copper track layout and component mounting plan of the PCB are given in **Figure 3**. Actually, 'components' is limited to a few wire links and connectors that enable the serial link to be 'tapped'. So, we think you will agree with us that this is a very simple circuit indeed.

Script language

COM-Watch has an extensive script language, which allows almost any problem to be pinpointed, even if it occurs sporadically. This overview lists the commands available for making scripts. Script files intended for COM-Watch have the extension .cws.

Syntax

[] means optional

| means 'or'

drive: specifies disk station

path: specifies location

filename: self-evident

string: a series of characters enclosed in brackets

control: specifies the status of control signals the system has to wait for. The following options are available:

CTS = on | off

DSR = on | off

RI = on | off

CD = on | off

BRK = on | off

FE = on | off

PE = on | off

OE = on | off

Description of script commands

CAPTURE = on | off

Determines whether or not data from connected equipment has to be stored in a buffer, for later analysis.

CHDIR [*drive:*]*path*

Change current directory

EXIT [*PROGRAM*]

Leave the script program and return to analysis mode. If the parameter PROGRAM is added, the program returns to DOS mode.

GOTO *label*

Only the first 8 characters of a label are used to distinguish it from other labels. Up to 100 labels may be used in a script.

KEYBOARD on | off

Switch the keyboard on or off. This allows you to simulate a terminal.

MATCHCASE on | off

Determine if the script language is case sensitive or not. The default is MATCHCASE off.

SEND *string*

Transmit a text line to the connected peripheral, for example, a modem initialization string.

SEND FILE [*drive:*]*[path]filename*

Transmit the indicated file to the peripheral.

SET CHANn = COM*m* | off

Select a communication port. For parameters *n* and *m* the values 1 through 4 are available. A COM-Watch channel is switched off with off.

SET CHANn IDENT = *string*

Defines an identification string for a channel. Parameter *n* is either 1 or 2. Only the first 10 characters of a string are used by the program.

The default identifiers are

CHAN1 IDENT = Channel #1

CHAN2 IDENT = Channel #2

SET COMm = *baud* [, *parity* [, *data* [, *stop*]]] COM*m*

Specifies the COM port, with *m* between 1 and 4.

baud

Specifies the communication speed; available values are 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200.

data

Specifies 7 or 8 databits

stop

Specifies the number of stop bits.

The default values are copied from DOS.

SET DISKSTREAMING on [, *drive:*]*[path]filename* | off

If no extension is given with the filename, the program chooses the extension .cwsd.

SET FORMAT *mnem* | *ascii* | *hex* | *dec* | *oct* | *ebcdic*

This command changes the current format into the selected format.

SET RINGBUFFER on | off

Switch the ring buffer on or off.

SET SENDDELAY *milliseconds*

This is used to set the length of the pause between consecutive characters in the transmission of data. The available range is 0 to 100 ms. The default is no delay.

SET TIMESTAMPING *abs* | *rel* | *date* | off

This command is used to select the operation of the time stamping system.

SET VIEW *split* | *mixed* | *channel 1* | *channel 2* | *event* | *active*

This command is used to set the current display mode.

WAIT COUNT_EVENTS *number*

Determines how many events must have occurred before execution of the script is continued. Maximum value: 1000.

WAIT DATA_EVENT [*CHANn*] *string* [*control*]

Interrupts the execution of a script until the stated data have been received.

[*CHANn*]

This may be used to optionally define the channel. If no channel number is indicated, both channels are monitored.

string

Specifies the string to wait for.

control

Specifies the status of the control signals to wait for.

WAIT DELAY *hundredths_of_seconds*

Execution of the script is halted and resumed after the specified delay.

WAIT KEYSTROKE *string*

Execution of the script is halted and not resumed until the specified text has been entered via the keyboard.

WAIT TIME_EVENT [*month-day-year*] *hours:minutes:seconds*

This specifies the date and time when script execution is to resume. Month, day and year are optional.

WHEN DATA_EVENT [*Chan*] *string* [*control*] [, *command*]

Indicates which action has to be taken if a specific situation arises.

CHANn

This command allows you to optionally specify the channel. If no number is supplied, both channels are monitored.

string
This specifies the text to wait for.

control
Specifies the status of the control signals to wait for. This is an optional addition.

command
Defines the action to be taken if a condition is satisfied.

WHEN KEYSTROKE *string[,command]*
This command specifies the action to be taken after a string has been entered.

string
Specifies the text to wait for.

command
Specifies the action to be taken when the condition is satisfied.

WHEN TIME_EVENT [*month-day-year*] *hours:minutes;seconds* [*command*]
This command specifies the action undertaken when a certain time is reached.

month
Specifies the month; valid numbers are 1 through 12.

day
Specifies the day; valid numbers are 1 through 31.

year
Specifies the year; valid numbers are 1980 through 2099.

hours:minutes;seconds
Specifies the exact time at which the script is to be executed.

hours
Specifies hours, valid numbers are 0 through 23.

minutes
Specifies minutes, valid numbers are 0 through 59.

seconds
Specifies minutes, valid numbers are 0 through 59.

Command
This is the command to be executed when the indicated date and time are reached. If no command is entered, COM-Watch will switch off the previously defined WHEN DATA_EVENT with the same string.

The component mounting plan in **Figure 2a** shows the locations of the wire links if the interface is employed for analysing regular serial links (RS232). If you require support for a symmetrical serial bus like RS422, another configuration is in order. The relevant 'constellation' of the wire links is shown in **Figure 2b**. D25 and D9 connectors are fitted at both sides of the PCB. If you do not envisage ever needing one of the two, then simply omit the relevant connectors. If you only use the 9-pin version, the board may be made much smaller by omitting the 25-pin connectors.

The link to the PC (carrying the information employed for Channel 1), is made using two 9-way (male) connectors.

The software

COM-Watch is a program which runs under the DOS operating system. It is suitable for analysis of data traffic between a DTE and DCE. Fortunately, the fact that the DOS environment has been chosen for the program does not preclude its use under the Windows operating system. When the program is started, it simply claims the entire screen, temporarily hiding Windows.

Install the software on drive C: in a directory called ComWatch. If you want to use another drive, say, D:, then type

```
a:install d:
```

Any other drive number may be used, provided it exists.

After entering

```
[drive][path] Comwatch.exe
```

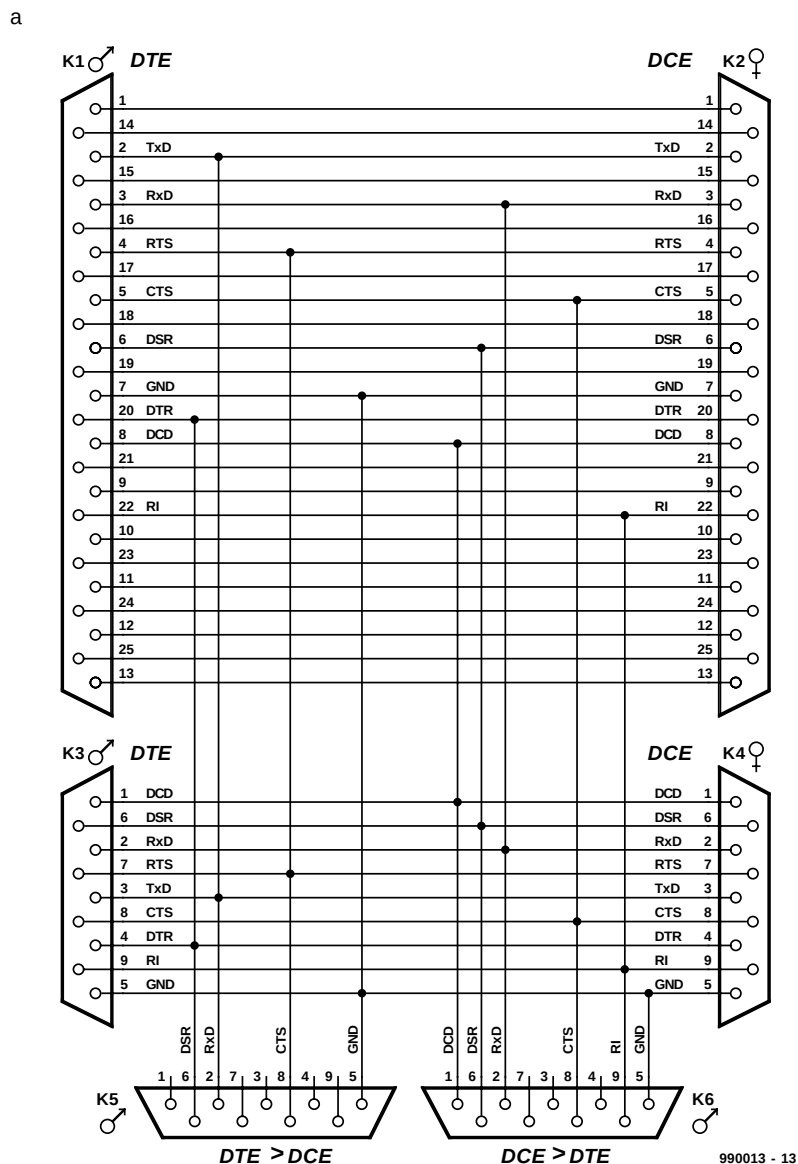


Figure 2a. Circuit diagram of the adaptor used for measurements in passive mode (RS232 version).

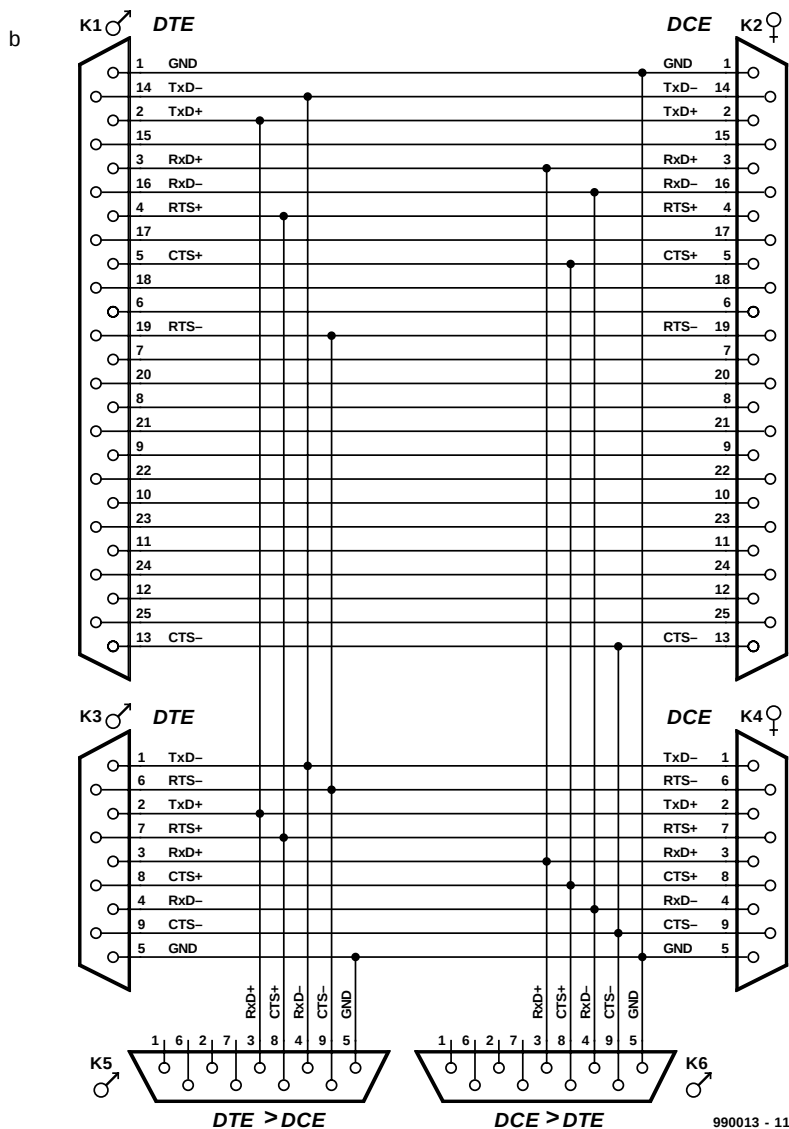
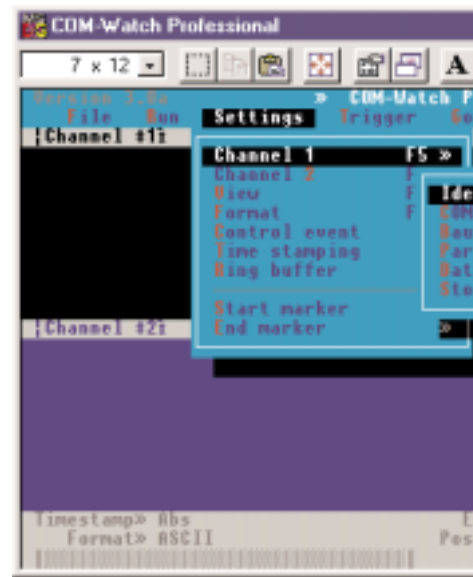


Figure 2b. Circuit diagram of the adaptor used for measurements in passive mode (RS422 version).



the program starts in full-screen mode. If the hardware has been connected up, the measurement session may start.

The screen shows quite a few things. Fortunately an interactive and context-sensitive Help function is available, so that everything will soon be clear. **Table 2** shows the functions assigned to various keys.

After connecting the hardware to the PC, you first have to configure the software. As you can see from the table, this is done with the aid of function keys F5 and F6. Once you have entered the right data, the program is ready for use.

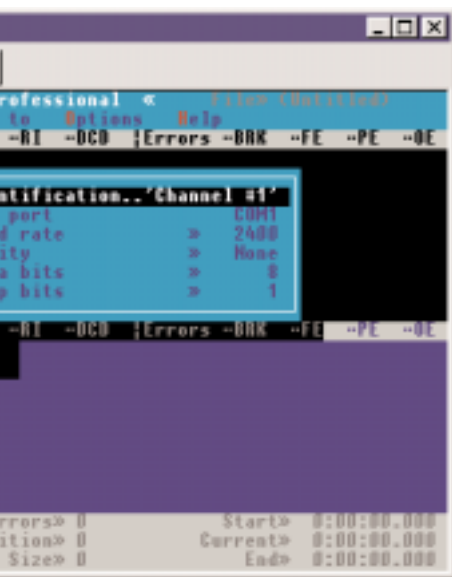
Table 1. Signal allocation on 25-pin and 9-pin RS232 connection.

D-25	D9	description		monitored	DTE-DCE
1		screening			
2	3	transmitted data	TxD	yes (#1)	DTE →
3	2	received data	RxD	yes (#2)	← DCE
4	7	request to send	RTS	yes (#1)	DTE →
5	8	clear to send	CTS	yes (#2)	← DCE
6	6	data set ready	DSR	yes (#2)	← DCE
7	5	massa	S.GND	-	
8	1	data carrier detect	DCD	yes (#2)	← DCE
9		reserved			
10		reserved			
11		free			
12		data carrier detect ch. 2		no	← DCE
13		clear to send ch. 2		no	← DCE
14		transmitted data ch. 2		no	DTE →
15		transmitted sign. elem. timing		no	DTE →
16		received data ch. 2		no	← DCE
17		received sig. elem. timing		no	← DCE
18		free			
19		request to send ch. 2		no	DTE →
20	4	data term. ready	DTR	yes (#1)	DTE →
21		free			
22	9	ring indicator	RI	yes (#2)	← DTE
23		data rate select		no	DTE →
24		transm. sign. elem. timing		no	DTE →
25					

Note:
signals processed via 1st COM port (#1) DTE → DCE
signals processed via 2nd COM port (#2) DTE ← DCE

Table 2. COM-Watch function keys

FKey	Normal	+ Ctrl
F1	Help	Go to position
F2	Save as...	Go to Time location
F3	Open ...	Find...
F4	Calculation	View/Activity
F5	Settings	View/ Channel 1 Channel 1
F6	Settings	View/ Channel 2 Channel 2
F7	View	View/Split
F8	Format	Format/ Mnemonic
F9	Capture Mode	Set Start marker to Current position
F10	Analyse mode	Set End marker to current position



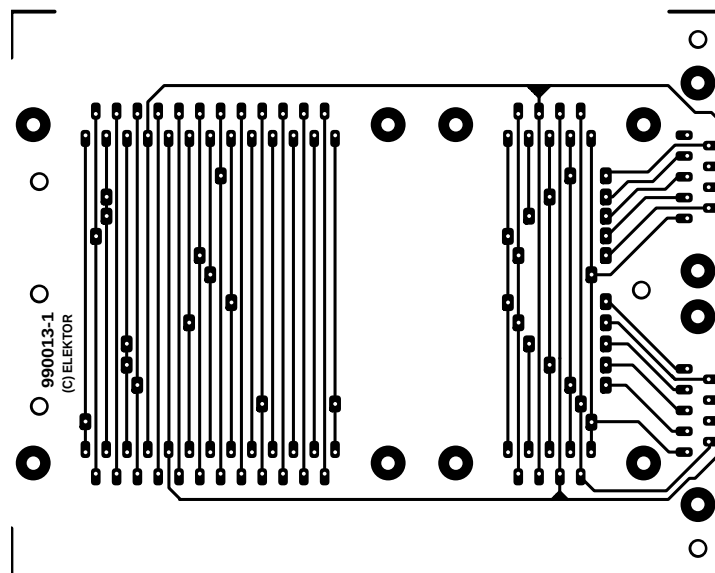
Finally we should mention that a more extensive version of the program is available. More information may be found on the CER Internet website at www.cer.com.

(990013-1)

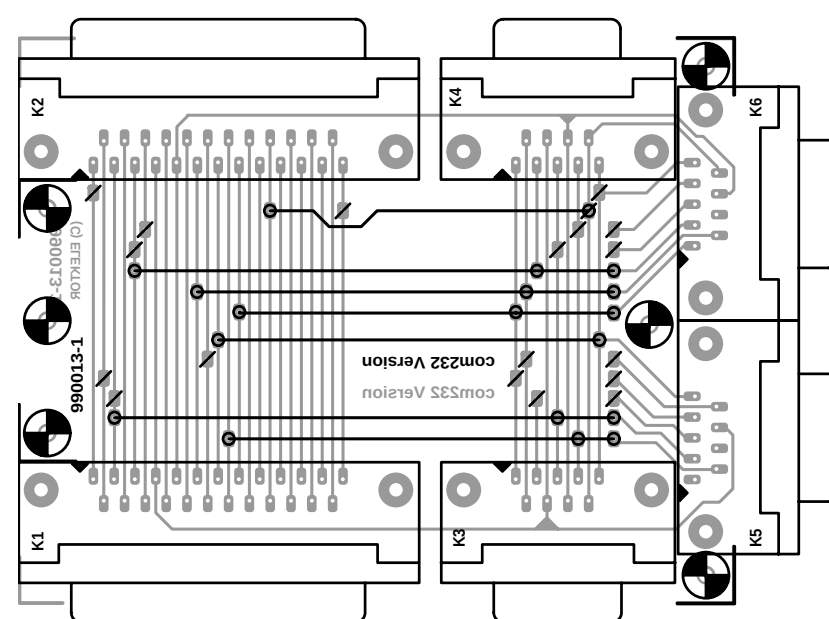
The PCB is available ready-made from the Publishers as order number **990013-1**.

The COM-Watch program is available on floppy disk (DOS) as order number **986035-1**.

+Alt	+Shift
Go to top	Go to Start marker
Go to Bottom	Go to End marker
Next match	Next error
Previous match	Previous error
Com port	Baudrate
Channel 1	Channel 1
Com port	Baudrate
Channel 2	Channel 2
View/Mixed	View/Events
Format/ASCII	Format/
	Hexadecimal
Set Start marker to	Set Start marker to
Top position	Entered position
Set End marker to bottom position	Set End marker to entered position



a



b

Figure 3. Copper track layout and component mounting plan of the adaptor board used for the passive monitoring system. Figure 4a is for the RS232 version, Figure 4b for the RS422 version.